SOFTWARE FRAMEWORK FOR PROGRESS TRACKING IN A CLASSROOM SETTING

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] The present application claims the benefit of U.S. Provisional Application No. 62/648,361, entitled "SOFTWARE FRAMEWORK FOR PROGRESS TRACKING IN A CLASSROOM SETTING," filed Mar. 26, 2018, the content of which is incorporated herein by reference in its entirety for all purposes.

FIELD

[0002] The described embodiments relate generally to a software framework configured to implement various techniques in a classroom setting. More particularly, the present embodiments relate to tracking and reporting information related to progress of completion of online assignments for a plurality of students.

BACKGROUND

[0003] Ever since the invention of audio visual (AV) devices such as televisions, projection systems, and tape decks, school districts and instructors have been incorporating materials into their lesson plans that utilize these devices to provide information to students in new and engaging ways. Instructors were not merely limited to textbooks, lectures, and written assignments. With the advent of the information age, the Internet has opened new horizons in the classroom. Instructors can now draw from a nearly unlimited resource of information in order to create interactive lesson plans that are engaging and productive.

[0004] However, the tools available to instructors when developing these lesson plans are limited. For example, a Wikipedia® article may provide useful background material on a particular subject. Nevertheless, the instructor may not have a good option to share the article with their students. The instructor could print out the article on paper hand-outs, which was the traditional way to distribute such materials to students. However, this method is wasteful and loses some of the interactive elements (e.g., animations, hyperlinks, etc.) of the article as presented online. Alternatively, the instructor could email a hyperlink for the article to each of their students to view on a computing device at home or provided by the school. While such methods allow the students to view the article within the proper context to interact with the interactive elements embedded within the article, there is no way for the instructor to track whether each student clicked on the hyperlink or even whether each student actually read through the entire article.

SUMMARY

[0005] This paper describes various embodiments that relate to a client-server architecture that enables progress tracking related to assignments generated by an instructor. A hand-out can include attachments that specify placeholders for hand-ins and/or activities to be performed by the student as part of the assignment. Some activities can be performed using third-party applications that implement at least a portion of a software framework that causes the applications to generate progress tracking information provided to a daemon executing in the background of a client device. The daemon and/or a client application interface with one or more network services to enable an instructor to utilize the progress tracking information. The network services can include a hand-out service utilized to generate hand-outs assigned to a group of students. The network services can also include a progress pipeline including a number of services configured to process progress tracking information received from a plurality of client devices.

[0006] In some embodiments, a client device includes a processor, a memory, and a display. The memory can include an operating system, a client application, a daemon, and one or more third-party applications, such applications including at least one progress application that implements at least a portion of a software framework. The software framework enables the progress application to generate progress tracking information related to one or more contexts the progress application provides to the client application and/or the daemon. A context generally defines a structure of at least a portion of the progress application such that the client application can define activities that can be completed by a student within the progress application.

[0007] In some embodiments, the daemon is executed as a background process and is configured to receive progress tracking information from one or more applications, filter the progress tracking information, store the progress tracking information in a remote database, generate metadata associated with the progress tracking information, and transmit the metadata to at least one service. The daemon can also implement a syncing mechanism that coordinates versions of a resource stored locally on the client device with versions stored in a remote database. The syncing mechanism enables the resource to be accessed on two or more client devices.

[0008] In some embodiments, the software framework causes the progress applications to generate API calls transmitted to the daemon that include progress tracking information. The daemon can be configured to filter the progress tracking information based on whether progress tracking is enabled or disabled for the one or more progress applications. Alternately, the daemon can be configured to filter the progress tracking information based on a list of active contexts.

[0009] In some embodiments, the remote database is a network-based storage service apportioned into a plurality of zones storing different data, each zone limited in scope to any combination of at least one of an organization identifier, a class identifier, a user identifier, a context identifier, or a hand-out identifier. Metadata corresponding to the progress tracking information is generated by the daemon and transmitted to a progress pipeline. The metadata includes a reference to the progress tracking information in the remote database as well as at least one of a class identifier, a hand-out identifier, a user identifier, or a context identifier.

[0010] In some embodiments, the progress pipeline includes a number of services structured in a pipeline architecture. The progress pipeline can include an on-ramp service configured to: receive metadata corresponding to progress tracking information; de-reference a user identifier included in the metadata by replacing the user identifier with a directory service identifier or an internal identifier; determine whether progress tracking is enabled or disabled for one of the user identifier, the directory service identifier, or the internal identifier; and filter the metadata based on the determining. The progress pipeline can include an enrichment service configured to: enrich the metadata with at least one of an organization identifier, a directory service identi-